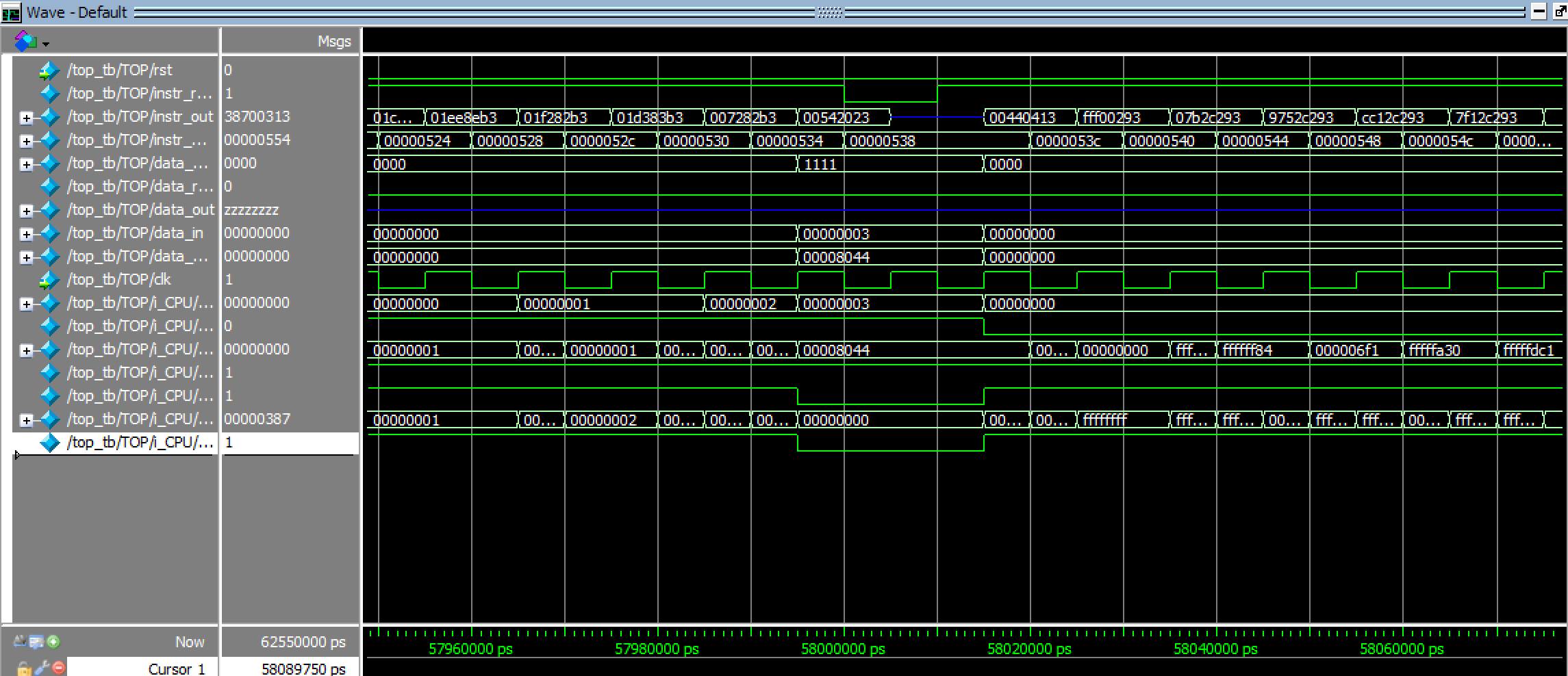
**Computer Organization 2019**

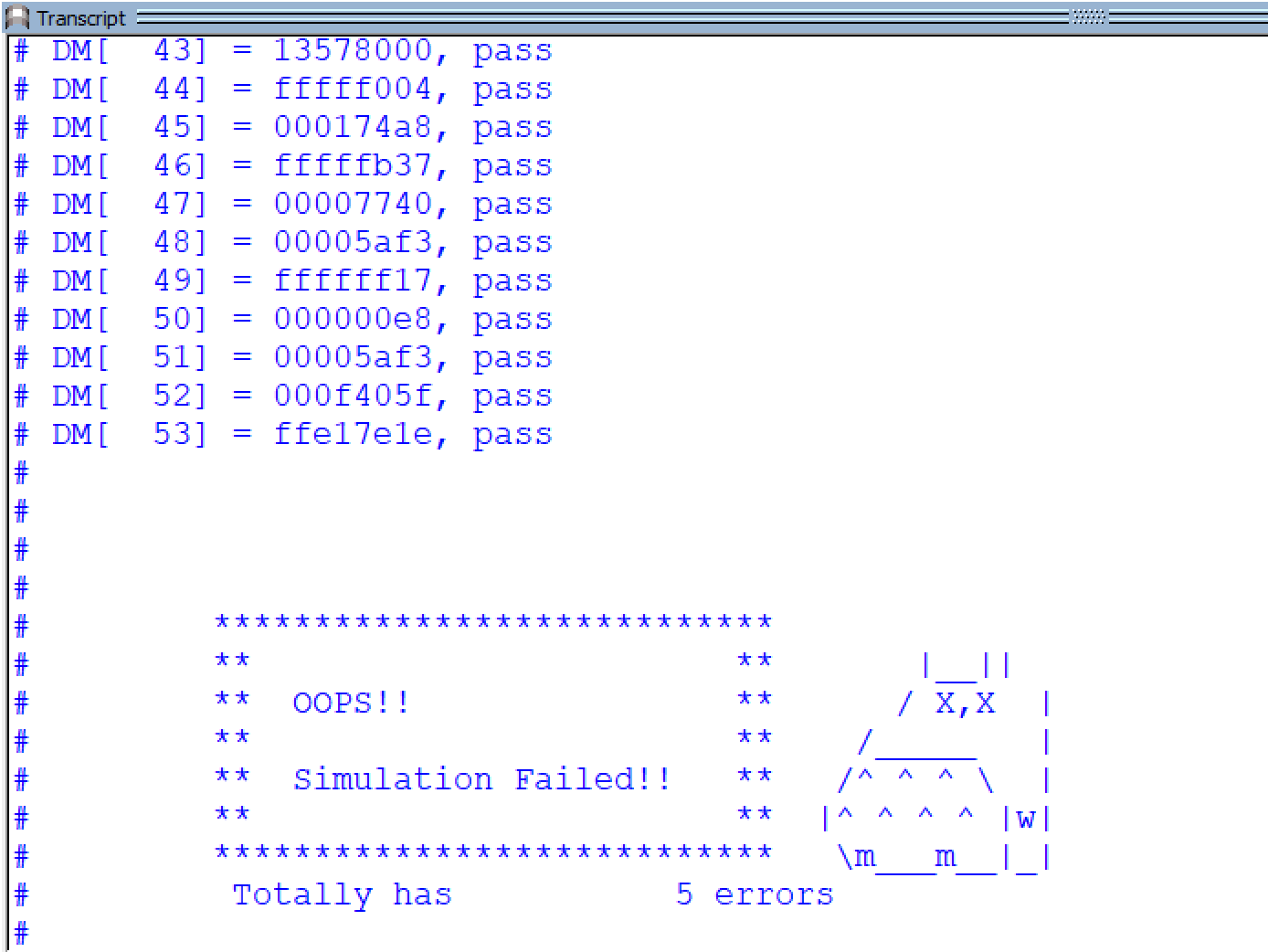
**HOMEWORK 4**

系級: 資訊系二 學號: H54086135 姓名:陳以新

**實驗結果圖:**

(波形圖及模擬完成截圖)



****

**程式運作流程:**

(簡單說明波形變化的意義)

根據作業要求，大致有以下訊號：

instr\_read : 訊號為1則讀instruction進來，否則不讀

instr\_addr : 表示從IM得到的指令位置

instr\_out : 從IM獲得的指令

data\_read : 是否要從DM讀data

data\_write : 決定要把data寫在哪些byte上

data\_addr : 在DM裡data的位置

data\_in : 是否要把data寫到DM

data\_out : 從DM獲得的data

因為認為電路稍複雜，拆成多個module寫好控制訊號較直觀且較無需擔心delay的問題，故還有許多wire例如imm做sign-extension的結果，或alu運算的結果等，在此就不加以贅述。

**心得**

(請寫下完成本次作業的心得、學到哪些東西、困難點的部分。)

從前就聽聞這份作業很難，且硬體描述語言一直是我的罩門，寫個紅綠燈都有點怕怕的，從不認為自己能寫完這份作業而一拖再拖，想等學霸朋友們寫完再動工，事實證明悲劇就是這樣發生的。一開始完全不知從何著手，看著流程圖想著指令們，理解題意就花了不少時間，到最後硬著頭皮參考github他人寫好的MIPS single-cycle CPU，拆成很多module，配合著流程圖按圖施工。從無到有，dubg時從初始化失敗讀到xxxxx再到讀到00000，實力本來就不如人的我debug每de玩一個bug都格外興奮，雖然最後沒辦法完成全部的指令，對照一下應該是load的指令出事，但也不曉得自己為什麼store能好好的運作，卻load了不知道從哪裡來的data。正逢疫情升溫，完全不必擔心自己在家裡沒事做，非常感激老師的用心良苦，讓一份又一份的作業陪伴著在家認真防疫的自己。如果自己早點動工的話應該是能de完這個bug。所以這次作業不僅讓我更熟悉CPU的運作及各指令的流程，更讓我深切明白作業要及早動工，絕不能總想著自己做不到而一再拖延，打一行是一行，答案終能漸漸明朗。